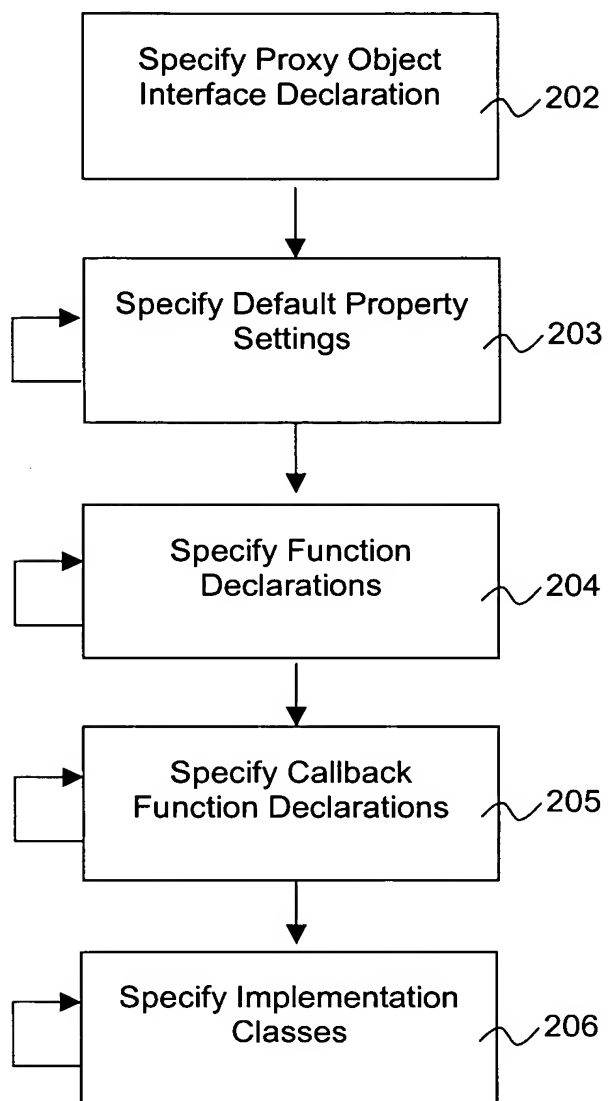


Figure 1



**Figure 2**

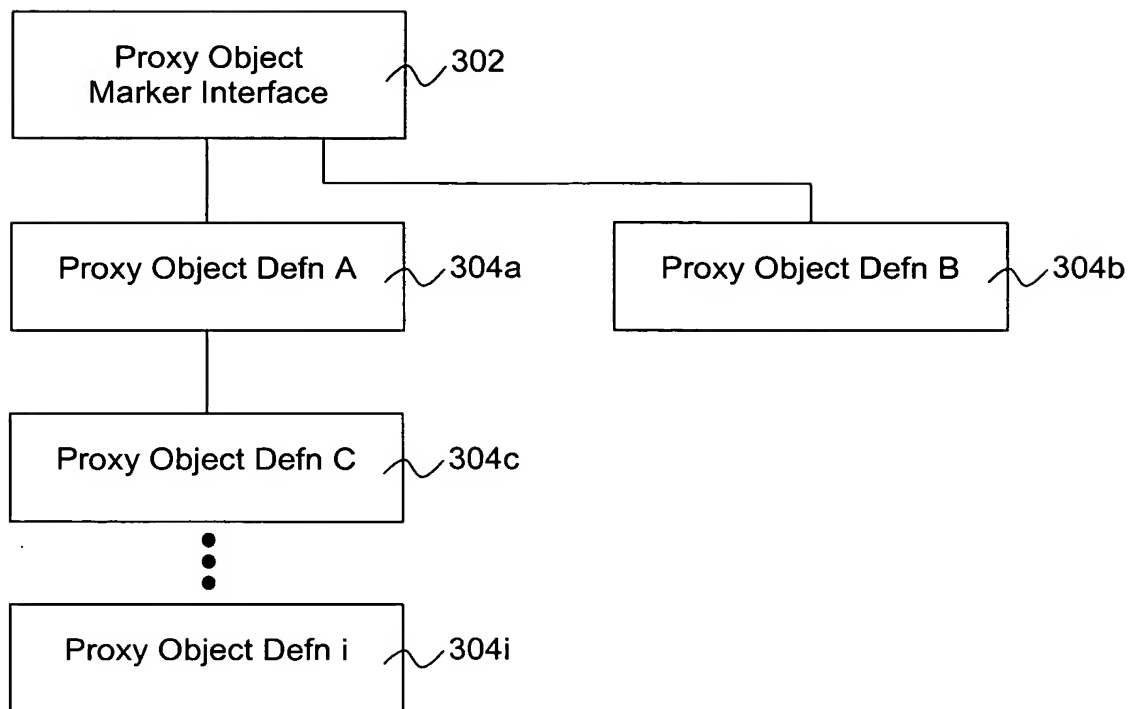


Figure 3

```

package com.bea.jws;                                400

/**
 * @implementation ~ 410
 * runtime ="com.bea.jws.private.TimerImpl" ~ 412
 * compiletime ="com.bea.jws.private.TimerValidator" ~ 414
 * designtime ="com.bea.jws.private.TimerDesigner" ~ 416
 */
interface Timer extends com.bea.jws.ProxyObject ~ 402
{
    void setTimeoutn(int milliseconds); ~ 404a
    void setTimeoutAt(java.util.Date date); ~ 404b

    java.util.Date getTimeoutAt();
    long getTimeoutIn();

    long clearTimeout();

    static public interface Callback
    {
        void onTimeout(); ~ 404c
    }
}

```

Figure 4

|  |            |
|--|------------|
| Proxy Object Implementation                                | <u>110</u> |
| Built-in Functions [F <sub>1</sub> , F <sub>2</sub> , ...] | <u>111</u> |
| Built-in Callbacks [C <sub>1</sub> , C <sub>2</sub> , ...] | <u>112</u> |
| Build Interface Functions                                  | <u>118</u> |
| - Get Property Syntax ( ) ~ 502                            |            |
| - Validate Class Properties ( ) ~ 504                      |            |
| - Validate Field Properties ( ) ~ 506                      |            |
| Resource Interface Functions                               | <u>116</u> |
| - Acquire Resource ( ) ~ 512                               |            |
| - Release Resource ( ) ~ 514                               |            |
| Extension Interface Functions                              | <u>114</u> |
| - Invoke ~ 516   |            |

Figure 5

```

<syntax>
  <property name="sql" onMethod="yes"> ~ 602a
  <attribute name="statement" required="yes"/> ~ 604a
  <attribute name="maxCount"> ~ 604b
  <type><integer/></type> ~ 606
  <default>+infinity</default> ~ 608
  </attribute>
  <attribute name="returnType"/> ~ 604c
  </property>
  <property name="pool" onDecl="optional" onDef="optional" onFunc="optional"> ~ 602b
  <attribute name="name" required="yes"/>
  </property>
</syntax>

```

Figure 6

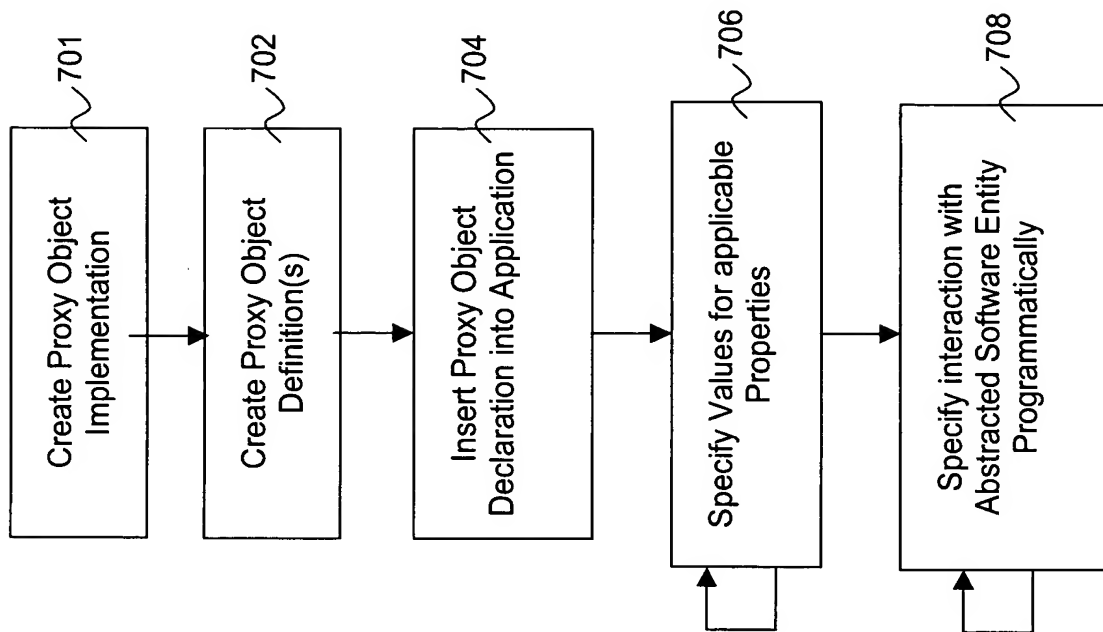


Figure 7

```
/**  
 * @timer  
 */  
com.bea.jws.Timer theTimer; ~ 802
```

**Figure 8a**

```
/**  
 * @timer timeoutIn="30 sec" ~ 804  
 */  
com.bea.jws.Timer theTimer; ~ 802
```

**Figure 8b**

```
void theTimer_onTimeout(long t) ~ 806  
{  
    System.out.println("Timeout at " + t);  
}
```

**Figure 8c**



Figure 9a

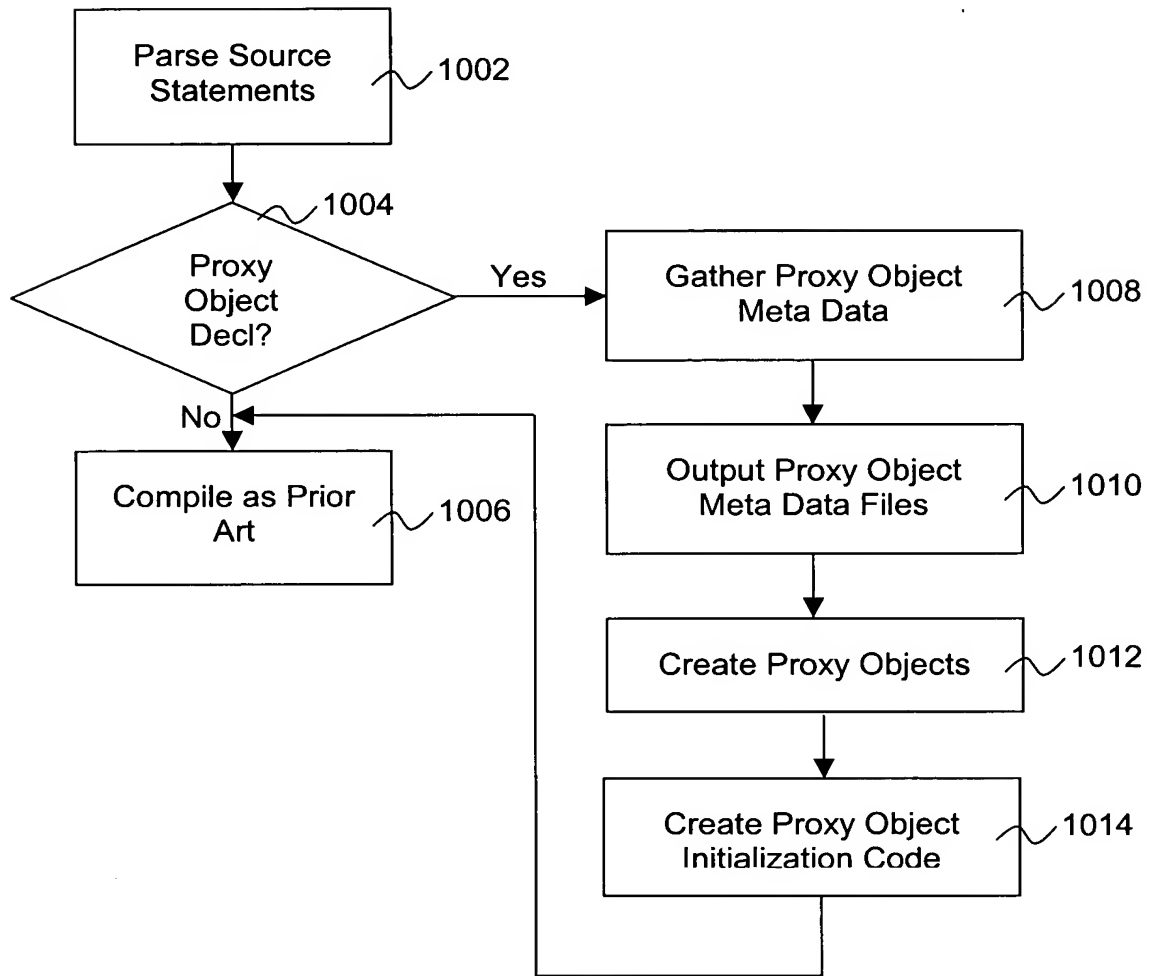
```
/**
 * StandardTimer.ctrl
 *
 * @timer timeoutIn="30 sec" ~ 904
 */
interface StandardTimer extends com.bea.jws.Timer ~ 902
{
}
```

```

EmployeeDB.ctrl ~ 920
*/
public interface EmployeeDB extends com.bea.jws.Database
{
    public class EmployeeRecord
    {
        String name;
        long salary;
    }
} ~ 922
~ 924
```

Figure 9b

```
/**
 * Port level resource attributes, such as the SQL associated
 * with the port, parameter maps, etc.
 *
 * @sql statement="SELECT NAME, SALARY FROM EMPS WHERE ID = {id}" ~ 926
 */
EmployeeRecord getEmployeeData(int id) throws SQLException; ~ 928
}
```

**Figure 10a**

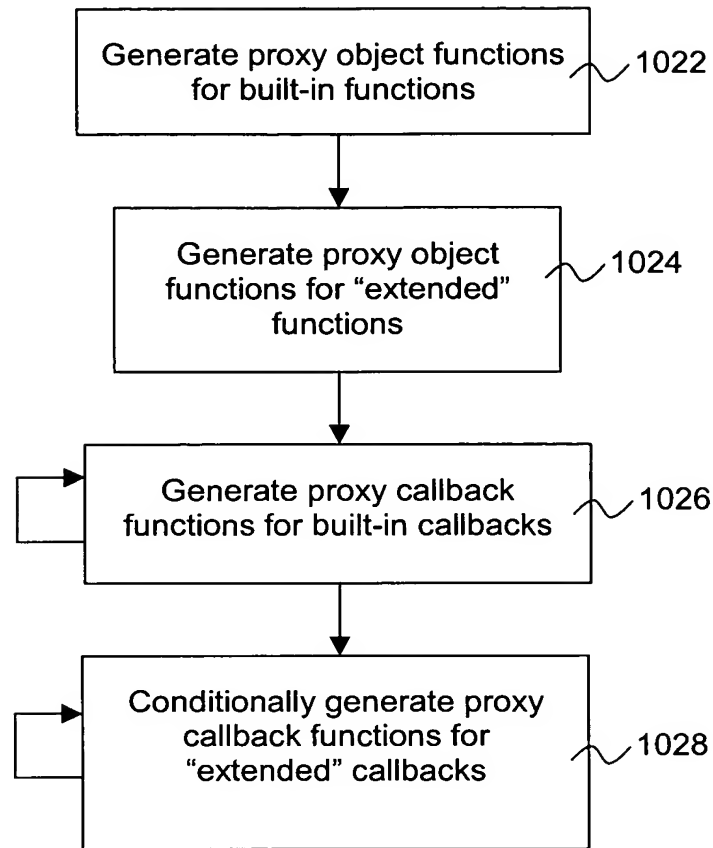


Figure 10b

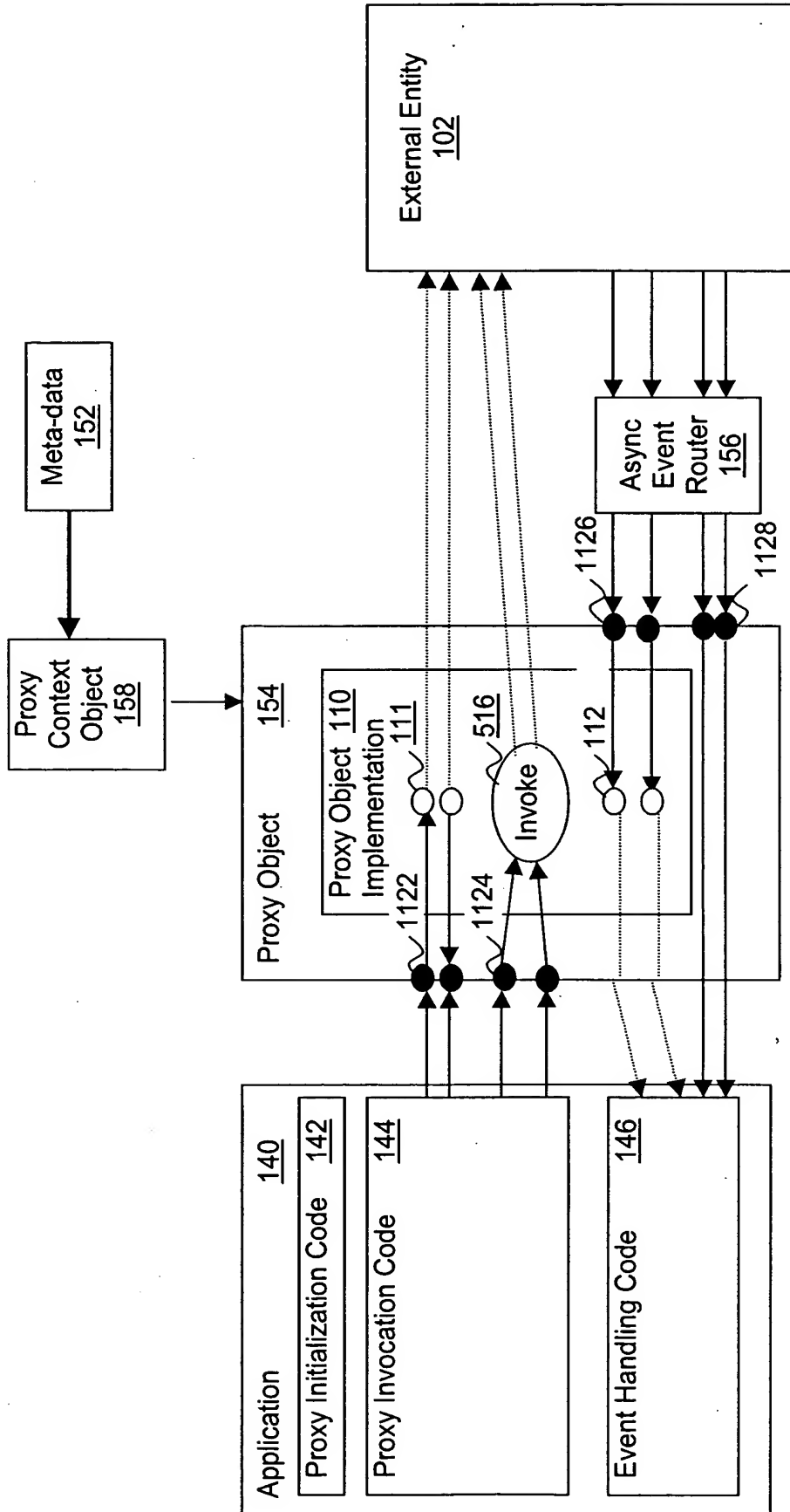


Figure 11

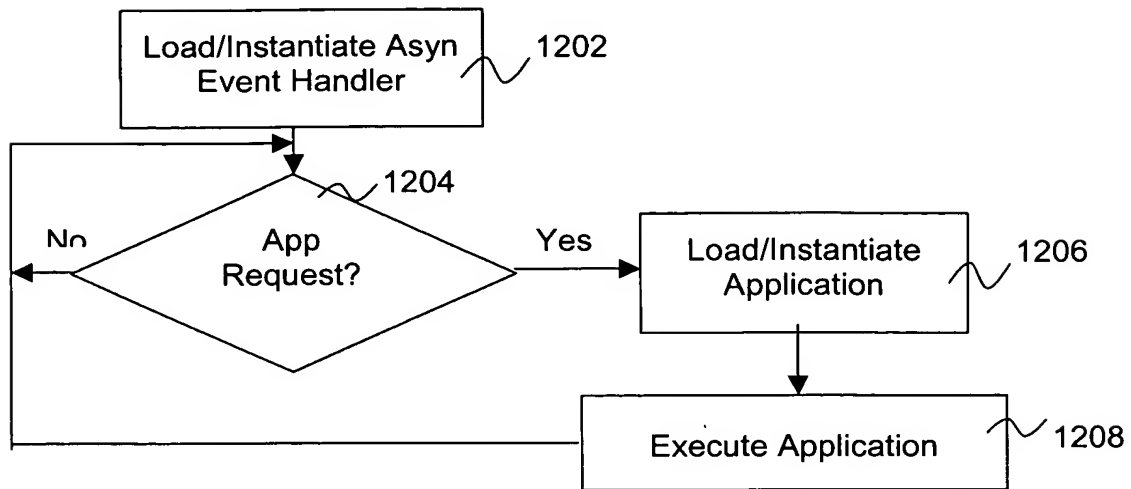


Figure 12a

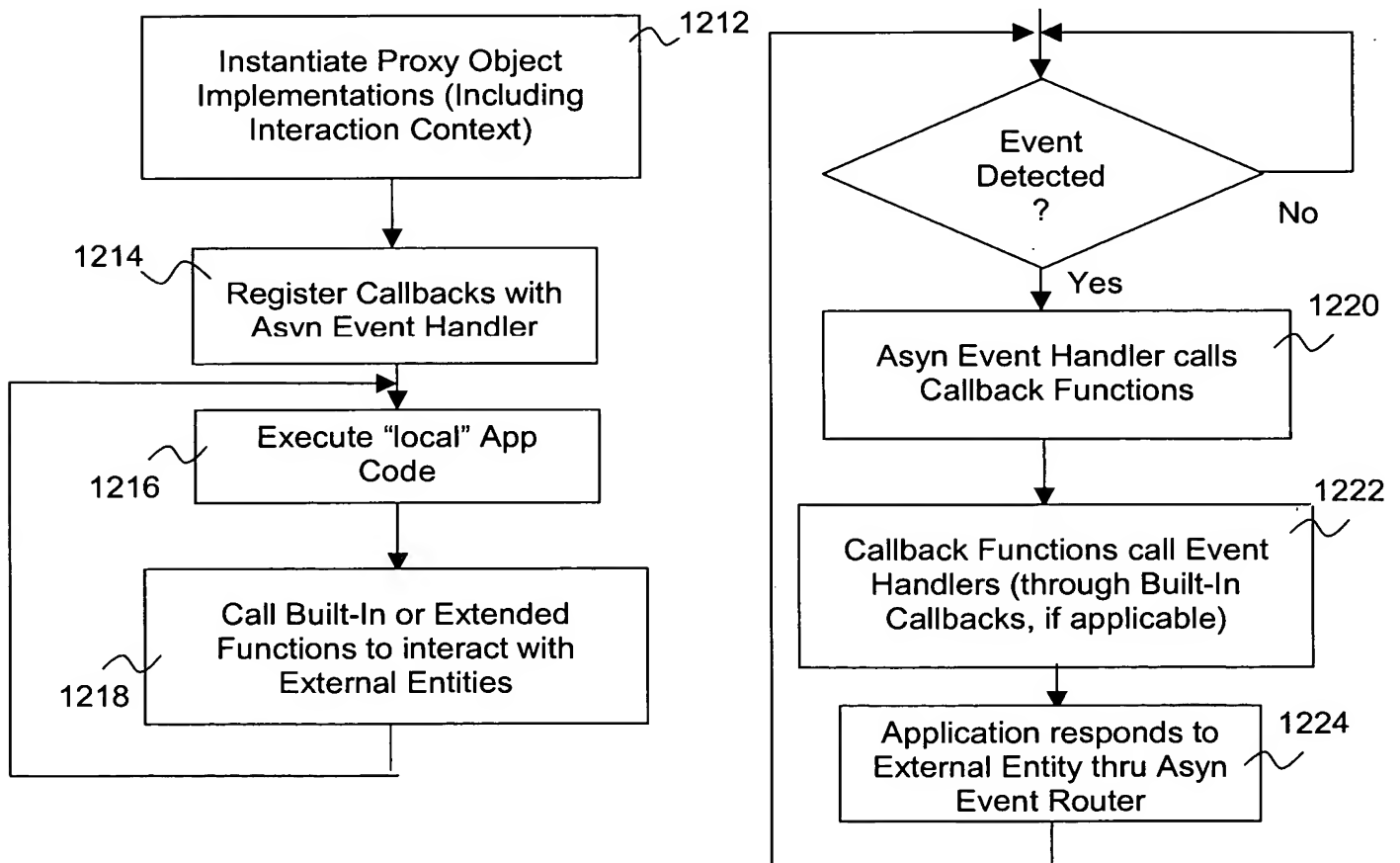


Figure 12b

**Figure 13a**

```
/**
 * @jws:control
 * @jws:timer timeout="1h"
 */
TimerFactory manyTimers;
```

**Figure 13b**

```
Timer t = manyTimers.create();

// then you can just use the control, store it, or whatever.
t.start();

// For example, let's associate a name with the timer...
timerMap.put(t, "First Timer");
```

**Figure 13c**

```
void manyTimers_onTimeout(Timer t, long elapsed)
{
    // let's retrieve the remembered name associated with the timer
    String timerName = (String)timerMap.get(t);

    // and print it out
    System.out.println("Event received from " + timerName);
}
```